Water quality analysis

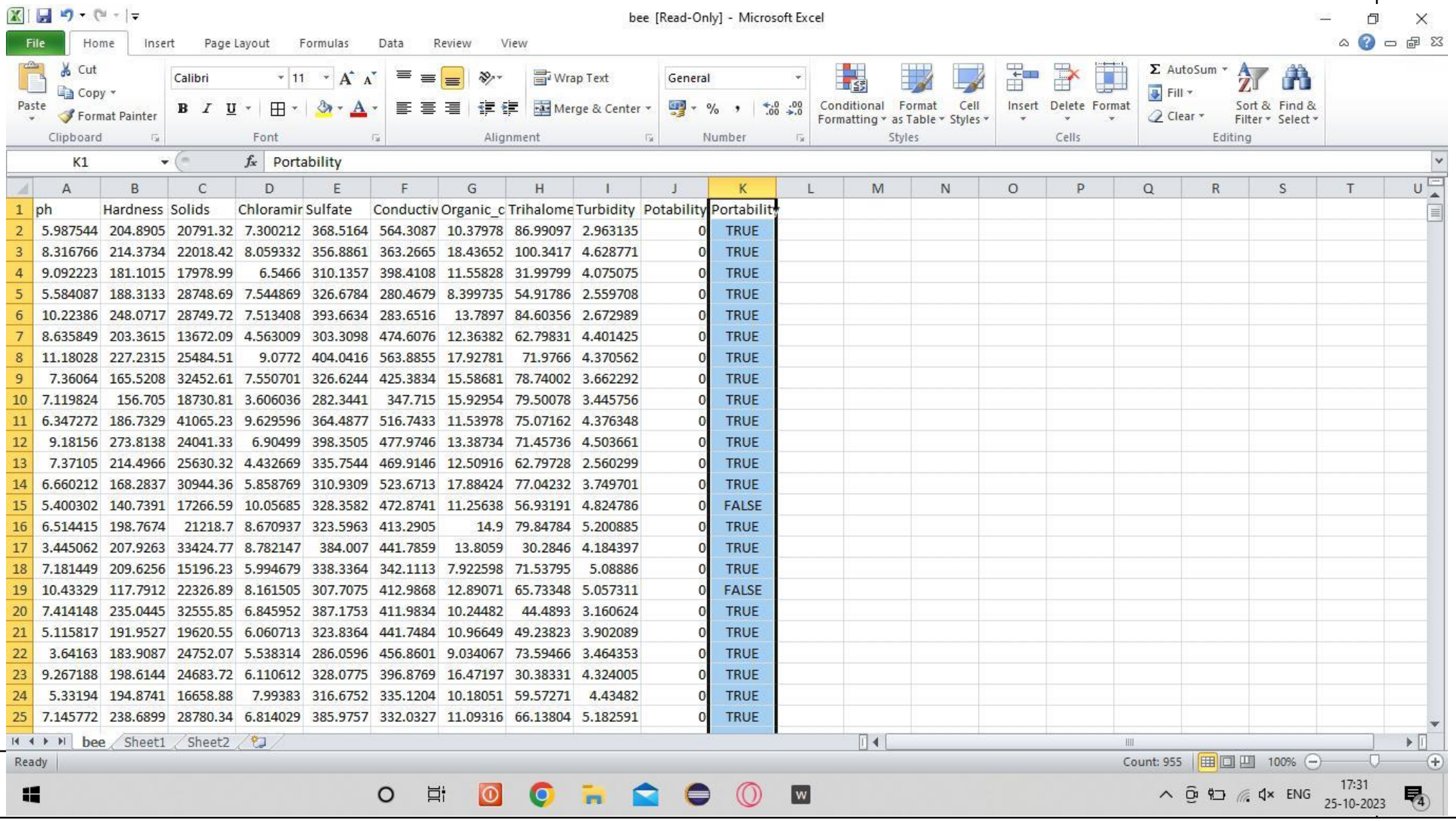
Phase4

Visualisation and predicating model:

- Loading the dataset
- Removing dummies
- Filling the missing values
- Update the new dataset
- Visualisation
- Creating a predictive model

Collection water quality analysis dataset:

Our dataset have 5003rows and 10 columns



	A	B	C	D	E	F	G	H	I	J	K
1	ph	Hardness	Solids	Chloramir	Sulfate	Conductiv	Organic_c	Trihalome	Turbidity	Potability	Portabilit
2	5.987544	204.8905	20791.32	7.300212	368.5164	564.3087	10.37978	86.99097	2.963135	0	TRUE
3	8.316766	214.3734	22018.42	8.059332	356.8861	363.2665	18.43652	100.3417	4.628771	0	TRUE
4	9.092223	181.1015	17978.99	6.5466	310.1357	398.4108	11.55828	31.99799	4.075075	0	TRUE
5	5.584087	188.3133	28748.69	7.544869	326.6784	280.4679	8.399735	54.91786	2.559708	0	TRUE
6	10.22386	248.0717	28749.72	7.513408	393.6634	283.6516	13.7897	84.60356	2.672989	0	TRUE
7	8.635849	203.3615	13672.09	4.563009	303.3098	474.6076	12.36382	62.79831	4.401425	0	TRUE
8	11.18028	227.2315	25484.51	9.0772	404.0416	563.8855	17.92781	71.9766	4.370562	0	TRUE
9	7.36064	165.5208	32452.61	7.550701	326.6244	425.3834	15.58681	78.74002	3.662292	0	TRUE
10	7.119824	156.705	18730.81	3.606036	282.3441	347.715	15.92954	79.50078	3.445756	0	TRUE
11	6.347272	186.7329	41065.23	9.629596	364.4877	516.7433	11.53978	75.07162	4.376348	0	TRUE
12	9.18156	273.8138	24041.33	6.90499	398.3505	477.9746	13.38734	71.45736	4.503661	0	TRUE
13	7.37105	214.4966	25630.32	4.432669	335.7544	469.9146	12.50916	62.79728	2.560299	0	TRUE
14	6.660212	168.2837	30944.36	5.858769	310.9309	523.6713	17.88424	77.04232	3.749701	0	TRUE
15	5.400302	140.7391	17266.59	10.05685	328.3582	472.8741	11.25638	56.93191	4.824786	0	FALSE
16	6.514415	198.7674	21218.7	8.670937	323.5963	413.2905	14.9	79.84784	5.200885	0	TRUE
17	3.445062	207.9263	33424.77	8.782147	384.007	441.7859	13.8059	30.2846	4.184397	0	TRUE
18	7.181449	209.6256	15196.23	5.994679	338.3364	342.1113	7.922598	71.53795	5.08886	0	TRUE
19	10.43329	117.7912	22326.89	8.161505	307.7075	412.9868	12.89071	65.73348	5.057311	0	FALSE
20	7.414148	235.0445	32555.85	6.845952	387.1753	411.9834	10.24482	44.4893	3.160624	0	TRUE
21	5.115817	191.9527	19620.55	6.060713	323.8364	441.7484	10.96649	49.23823	3.902089	0	TRUE
22	3.64163	183.9087	24752.07	5.538314	286.0596	456.8601	9.034067	73.59466	3.464353	0	TRUE
23	9.267188	198.6144	24683.72	6.110612	328.0775	396.8769	16.47197	30.38331	4.324005	0	TRUE
24	5.33194	194.8741	16658.88	7.99383	316.6752	335.1204	10.18051	59.57271	4.43482	0	TRUE
25	7.145772	238.6899	28780.34	6.814029	385.9757	332.0327	11.09316	66.13804	5.182591	0	TRUE

Missing values :

- Missing value can be treated using the following command

fillna()

IDLE Shell 3.11.6

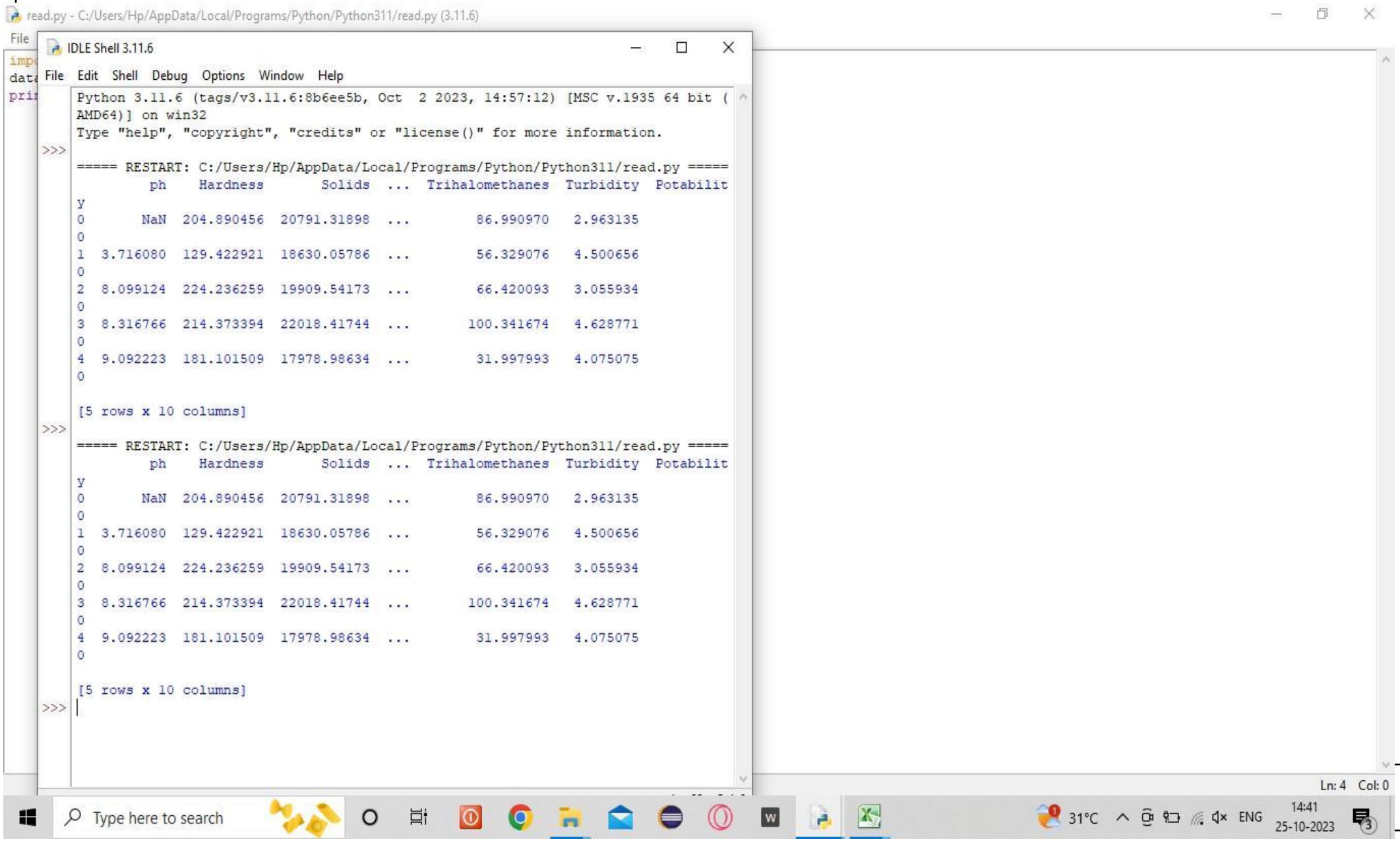
File Edit Shell Debug Options Window Help

```
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct  2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Hp/AppData/Local/Programs/Python/Python311/handling.py
          ph    Hardness        Solids  ...  Trihalomethanes  Turbidity  Potability
0        NaN  204.890456  20791.31898  ...        86.990970   2.963135           0
1   3.716080  129.422921  18630.05786  ...        56.329076   4.500656           0
2   8.099124  224.236259  19909.54173  ...        66.420093   3.055934           0
3   8.316766  214.373394  22018.41744  ...       100.341674   4.628771           0
4   9.092223  181.101509  17978.98634  ...        31.997993   4.075075           0

[5 rows x 10 columns]
>>>
```

Ln: 13 Col: 0

Loading our data:

- We can load our first five rows of data using the command
- import pandas as pd
- data=pd.read_csv(‘file_path’)
- Print(data.head())


read.py - C:/Users/Hp/AppData/Local/Programs/Python/Python311/read.py (3.11.6)
File
IDLE Shell 3.11.6
File Edit Shell Debug Options Window Help
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct 2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Users/Hp/AppData/Local/Programs/Python/Python311/read.py =====
ph Hardness Solids ... Trihalomethanes Turbidity Potability
0 NaN 204.890456 20791.31898 ... 86.990970 2.963135 0
1 3.716080 129.422921 18630.05786 ... 56.329076 4.500656 0
2 8.099124 224.236259 19909.54173 ... 66.420093 3.055934 0
3 8.316766 214.373394 22018.41744 ... 100.341674 4.628771 0
4 9.092223 181.101509 17978.98634 ... 31.997993 4.075075 0
[5 rows x 10 columns]
>>>
===== RESTART: C:/Users/Hp/AppData/Local/Programs/Python/Python311/read.py =====
ph Hardness Solids ... Trihalomethanes Turbidity Potability
0 NaN 204.890456 20791.31898 ... 86.990970 2.963135 0
1 3.716080 129.422921 18630.05786 ... 56.329076 4.500656 0
2 8.099124 224.236259 19909.54173 ... 66.420093 3.055934 0
3 8.316766 214.373394 22018.41744 ... 100.341674 4.628771 0
4 9.092223 181.101509 17978.98634 ... 31.997993 4.075075 0
[5 rows x 10 columns]
>>>
Ln: 4 Col: 0
Type here to search
31°C
ENG
14:41
25-10-2023

Encoded data:

- Since Python 3.0, strings are stored as Unicode, i.e. Each character in the string is represented by a code point. So, each string is just a sequence of Unicode code points. For efficient storage of these strings, the sequence of code points is converted into a set of bytes. The process is known as encoding.
- The code and output wiill be shown in the following slides.

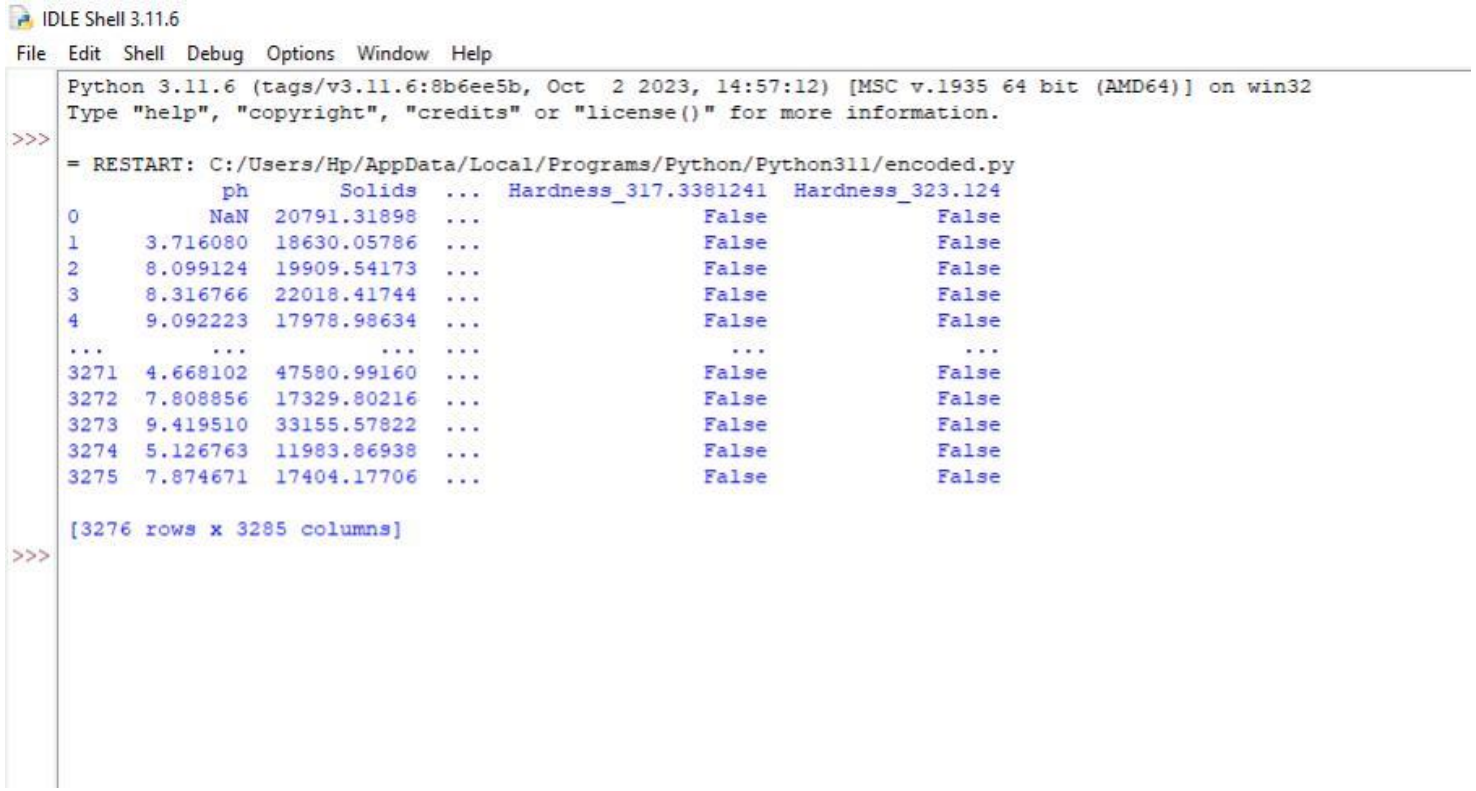
The code:

encoded.py - C:/Users/Hp/AppData/Local/Programs/Python/Python311/encoded.py (3.11.6)

File Edit Format Run Options Window Help

```
import pandas as pd
data=pd.read_csv(r'C:\Users\Hp\Documents\water.csv')

encoded_data=pd.get_dummies(data,columns=['Hardness'])

print(encoded_data)
```

Output


IDLE Shell 3.11.6
File Edit Shell Debug Options Window Help
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct 2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Hp/AppData/Local/Programs/Python/Python311/encoded.py
ph Solids ... Hardness_317.3381241 Hardness_323.124
0 NaN 20791.31898 ... False False
1 3.716080 18630.05786 ... False False
2 8.099124 19909.54173 ... False False
3 8.316766 22018.41744 ... False False
4 9.092223 17978.98634 ... False False
... ...
3271 4.668102 47580.99160 ... False False
3272 7.808856 17329.80216 ... False False
3273 9.419510 33155.57822 ... False False
3274 5.126763 11983.86938 ... False False
3275 7.874671 17404.17706 ... False False
[3276 rows x 3285 columns]
>>>

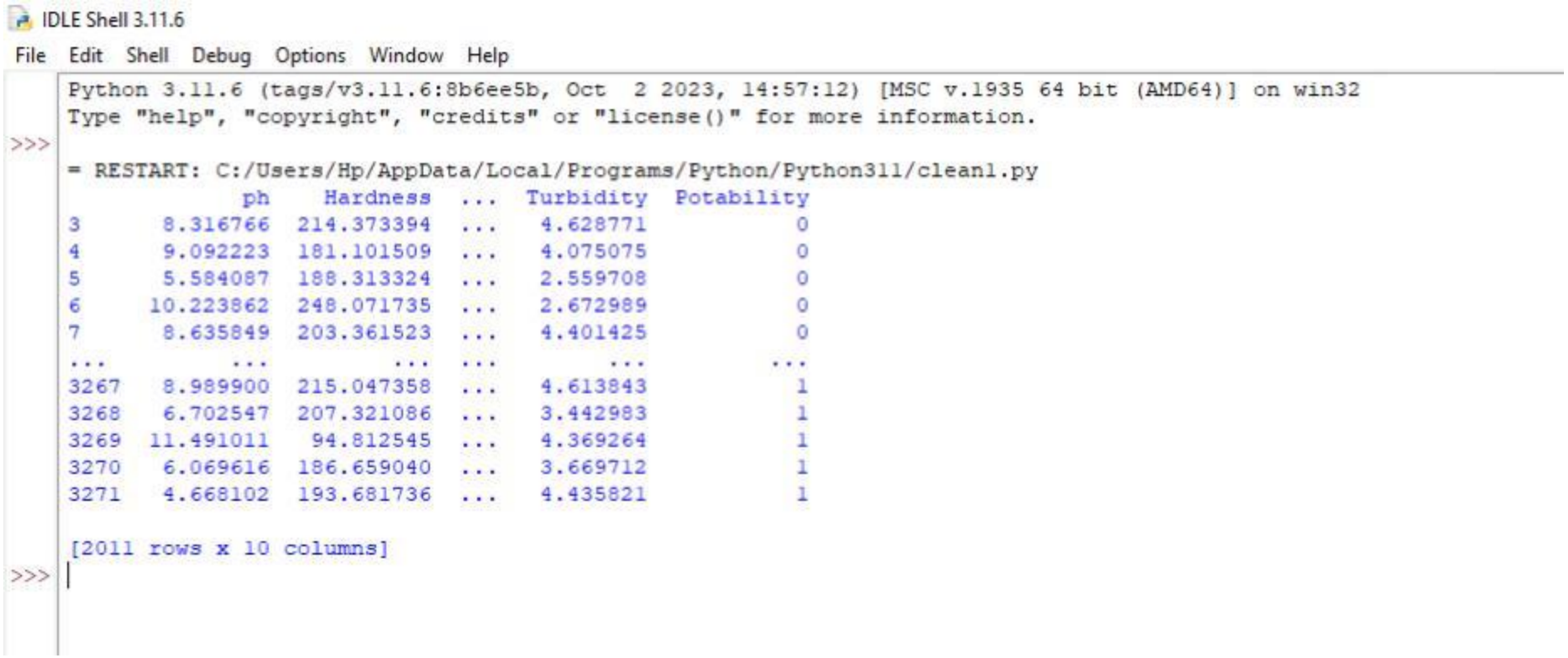
Dealing with outliers:

clean1.py - C:/Users/Hp/AppData/Local/Programs/Python/Python311/clean1.py (3.11.6)

File Edit Format Run Options Window Help

```
import pandas as pd
#load your dataset using this
data=pd.read_csv(r'C:\Users\Hp\Documents\water.csv')
#removes rows with any missing values
data=data.dropna()
#removing duplicate rows
data=data.drop_duplicates()
#dealing with outliers
def remove_outliers(data,column,z_threshold=3):
    z_scores=(data[column]-data[column].mean())/data[column].std()
    data=data[abs(z_scores)<z_theshold]
    return data
print(data)
```

OuTput:


IDLE Shell 3.11.6
File Edit Shell Debug Options Window Help
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct 2 2023, 14:57:12) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Hp/AppData/Local/Programs/Python/Python311/clean1.py
ph Hardness ... Turbidity Potability
3 8.316766 214.373394 ... 4.628771 0
4 9.092223 181.101509 ... 4.075075 0
5 5.584087 188.313324 ... 2.559708 0
6 10.223862 248.071735 ... 2.672989 0
7 8.635849 203.361523 ... 4.401425 0
... ...
3267 8.989900 215.047358 ... 4.613843 1
3268 6.702547 207.321086 ... 3.442983 1
3269 11.491011 94.812545 ... 4.369264 1
3270 6.069616 186.659040 ... 3.669712 1
3271 4.668102 193.681736 ... 4.435821 1
[2011 rows x 10 columns]
>>>

Update the new dataset:

- After performing all the preprocessing of dataset we can update our dataset as new dataset.We have performed missing values, outliers,filling the data using pandas packages in python.All of their required commands and output was shown.

Preprocessed data:

clean2.py - C:/Users/Hp/AppData/Local/Programs/Python/Python311/clean2.py (3.11.6)

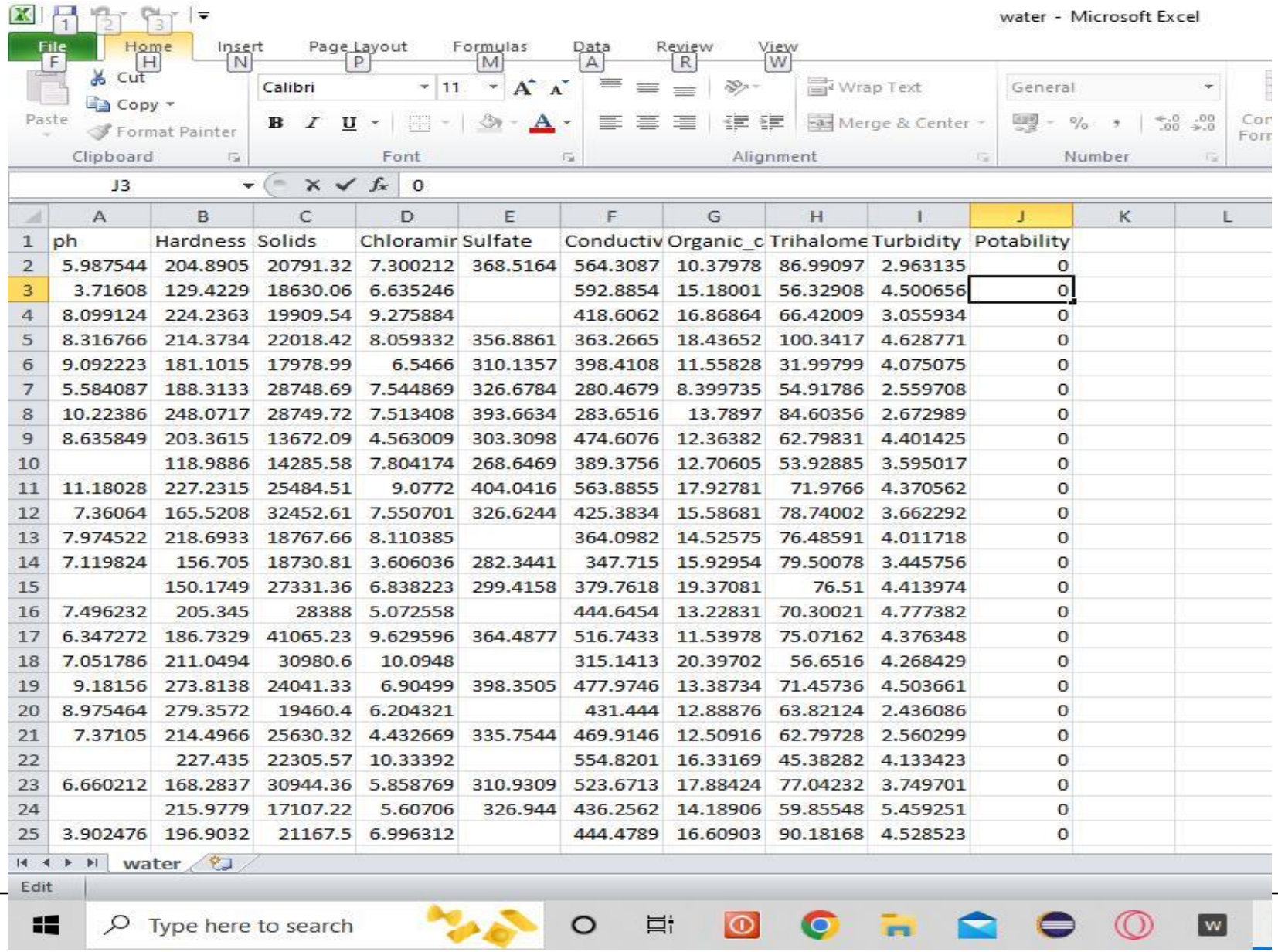
File Edit Format Run Options Window Help

```
import pandas as pd
#load your dataset using this
data=pd.read_csv(r'C:\Users\Hp\Documents\water.csv')
#removes rows with any missing values
data=data.dropna()
#removing duplicate rows
data=data.drop_duplicates()
#dealing with outliers
def remove_outliers(data,column,z_threshold=3):
    z_scores=(data[column]-data[column].mean())/data[column].std()
    data=data[abs(z_scores)<z_theshold]
    return data
print(data)



#After performing preprocessing of data we can save our dataset to a new csv file
data.to_csv('cleaned_dataset.csv',index=False)
```

IDLE Shell 3.11.6

File Edit Shell Debug Options Window Help

```
Python 3.11.6 (tags/v3.11.6:8b6ee5b, Oct  2 2023, 14:57:12) [MSC v.1935 64 bit
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/Hp/AppData/Local/Programs/Python/Python311/clean2.py
             ph    Hardness  ...  Turbidity  Potability
3      8.316766  214.373394  ...   4.628771           0
4      9.092223  181.101509  ...   4.075075           0
5      5.584087  188.313324  ...   2.559708           0
6     10.223862  248.071735  ...   2.672989           0
7      8.635849  203.361523  ...   4.401425           0
...         ...         ...  ...        ...         ...
3267   8.989900  215.047358  ...   4.613843           1
3268   6.702547  207.321086  ...   3.442983           1
3269  11.491011   94.812545  ...   4.369264           1
3270   6.069616  186.659040  ...   3.669712           1
3271   4.668102  193.681736  ...   4.435821           1

[2011 rows x 10 columns]
>>>
```



water - Microsoft Excel

File Home Insert Page Layout Formulas Data Review View

Cut Copy Format Painter Paste — Clipboard

Calibri 11 — Font

Wrap Text Merge & Center — Alignment

General — Number

J3 0

	A	B	C	D	E	F	G	H	I	J	K	L
1	ph	Hardness	Solids	Chloramir	Sulfate	Conductiv	Organic_c	Trihalome	Turbidity	Potability		
2	5.987544	204.8905	20791.32	7.300212	368.5164	564.3087	10.37978	86.99097	2.963135	0		
3	3.71608	129.4229	18630.06	6.635246		592.8854	15.18001	56.32908	4.500656	0		
4	8.099124	224.2363	19909.54	9.275884		418.6062	16.86864	66.42009	3.055934	0		
5	8.316766	214.3734	22018.42	8.059332	356.8861	363.2665	18.43652	100.3417	4.628771	0		
6	9.092223	181.1015	17978.99	6.5466	310.1357	398.4108	11.55828	31.99799	4.075075	0		
7	5.584087	188.3133	28748.69	7.544869	326.6784	280.4679	8.399735	54.91786	2.559708	0		
8	10.22386	248.0717	28749.72	7.513408	393.6634	283.6516	13.7897	84.60356	2.672989	0		
9	8.635849	203.3615	13672.09	4.563009	303.3098	474.6076	12.36382	62.79831	4.401425	0		
10		118.9886	14285.58	7.804174	268.6469	389.3756	12.70605	53.92885	3.595017	0		
11	11.18028	227.2315	25484.51	9.0772	404.0416	563.8855	17.92781	71.9766	4.370562	0		
12	7.36064	165.5208	32452.61	7.550701	326.6244	425.3834	15.58681	78.74002	3.662292	0		
13	7.974522	218.6933	18767.66	8.110385		364.0982	14.52575	76.48591	4.011718	0		
14	7.119824	156.705	18730.81	3.606036	282.3441	347.715	15.92954	79.50078	3.445756	0		
15		150.1749	27331.36	6.838223	299.4158	379.7618	19.37081	76.51	4.413974	0		
16	7.496232	205.345	28388	5.072558		444.6454	13.22831	70.30021	4.777382	0		
17	6.347272	186.7329	41065.23	9.629596	364.4877	516.7433	11.53978	75.07162	4.376348	0		
18	7.051786	211.0494	30980.6	10.0948		315.1413	20.39702	56.6516	4.268429	0		
19	9.18156	273.8138	24041.33	6.90499	398.3505	477.9746	13.38734	71.45736	4.503661	0		
20	8.975464	279.3572	19460.4	6.204321		431.444	12.88876	63.82124	2.436086	0		
21	7.37105	214.4966	25630.32	4.432669	335.7544	469.9146	12.50916	62.79728	2.560299	0		
22		227.435	22305.57	10.33392		554.8201	16.33169	45.38282	4.133423	0		
23	6.660212	168.2837	30944.36	5.858769	310.9309	523.6713	17.88424	77.04232	3.749701	0		
24		215.9779	17107.22	5.60706	326.944	436.2562	14.18906	59.85548	5.459251	0		
25	3.902476	196.9032	21167.5	6.996312		444.4789	16.60903	90.18168	4.528523	0		

water

Edit

Type here to search

Visualisation:

- Data visualization provides a good, organized pictorial representation of the data which makes it easier to understand, observe, analyze. In this tutorial, we will discuss how to visualize data using Python. Python provides various libraries that come with different features for visualizing data.

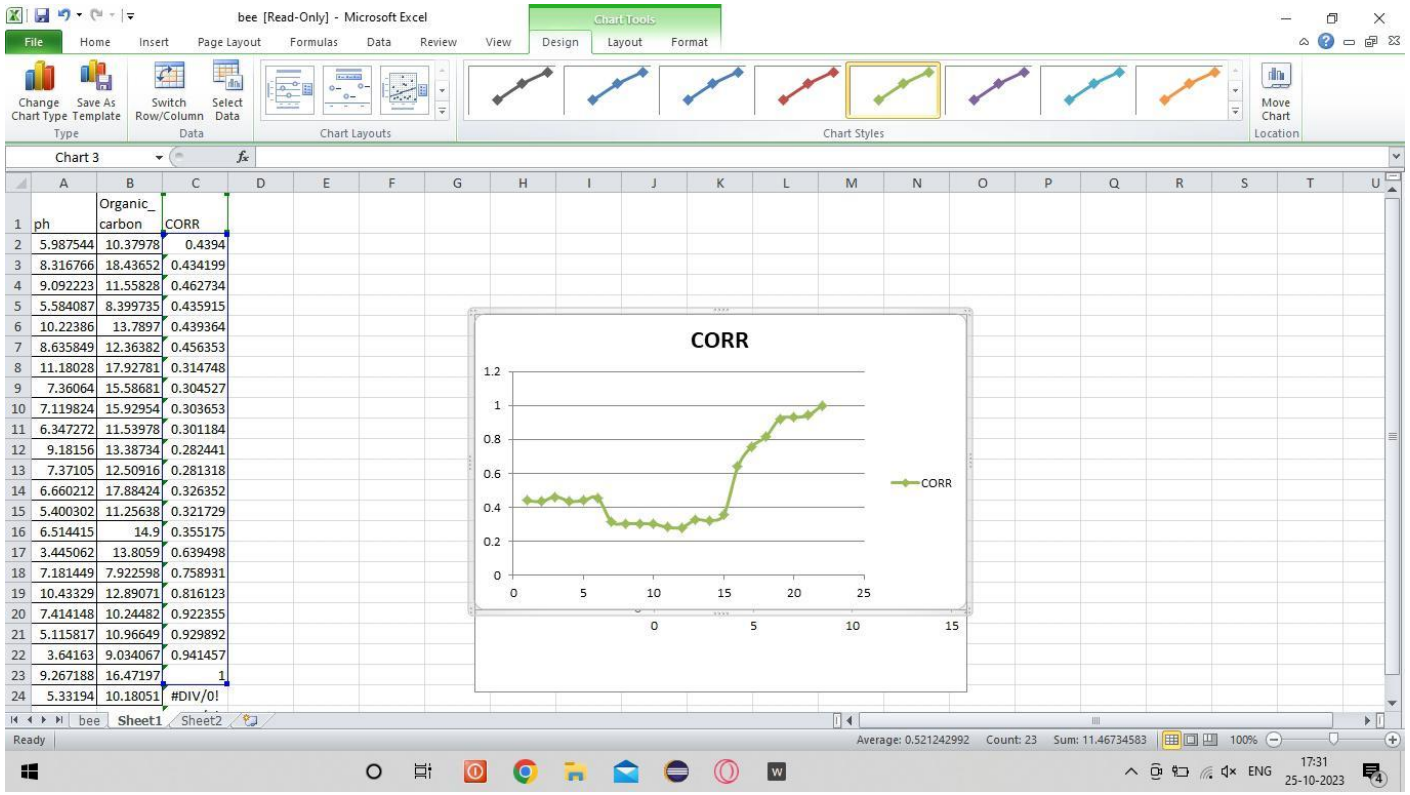
Five stages of visualisation:

- The five phases of visualization process: data gathering, processing, preparation, reduction and visual layout design. In recent years, a comparably fresh research field — information visualization has become commonly available for the researchers of all specialties.
- We have already performed the preprocessing of dataset.

Correlation:

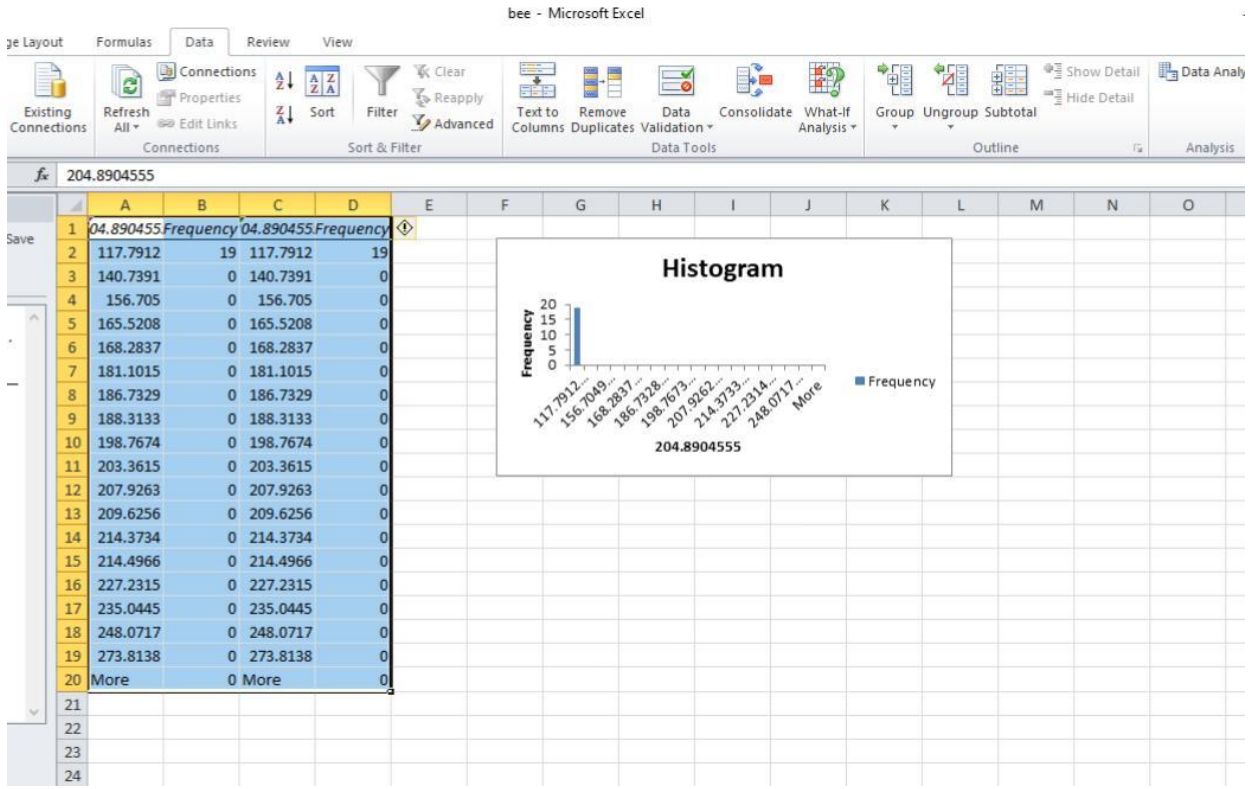
- Correlation summarizes the strength and direction of the linear (straight-line) association between two quantitative variables. Denoted by r, it takes values between -1 and +1. A positive value for r indicates a positive association, and a negative value for r indicates a negative association.

Correlation of data:



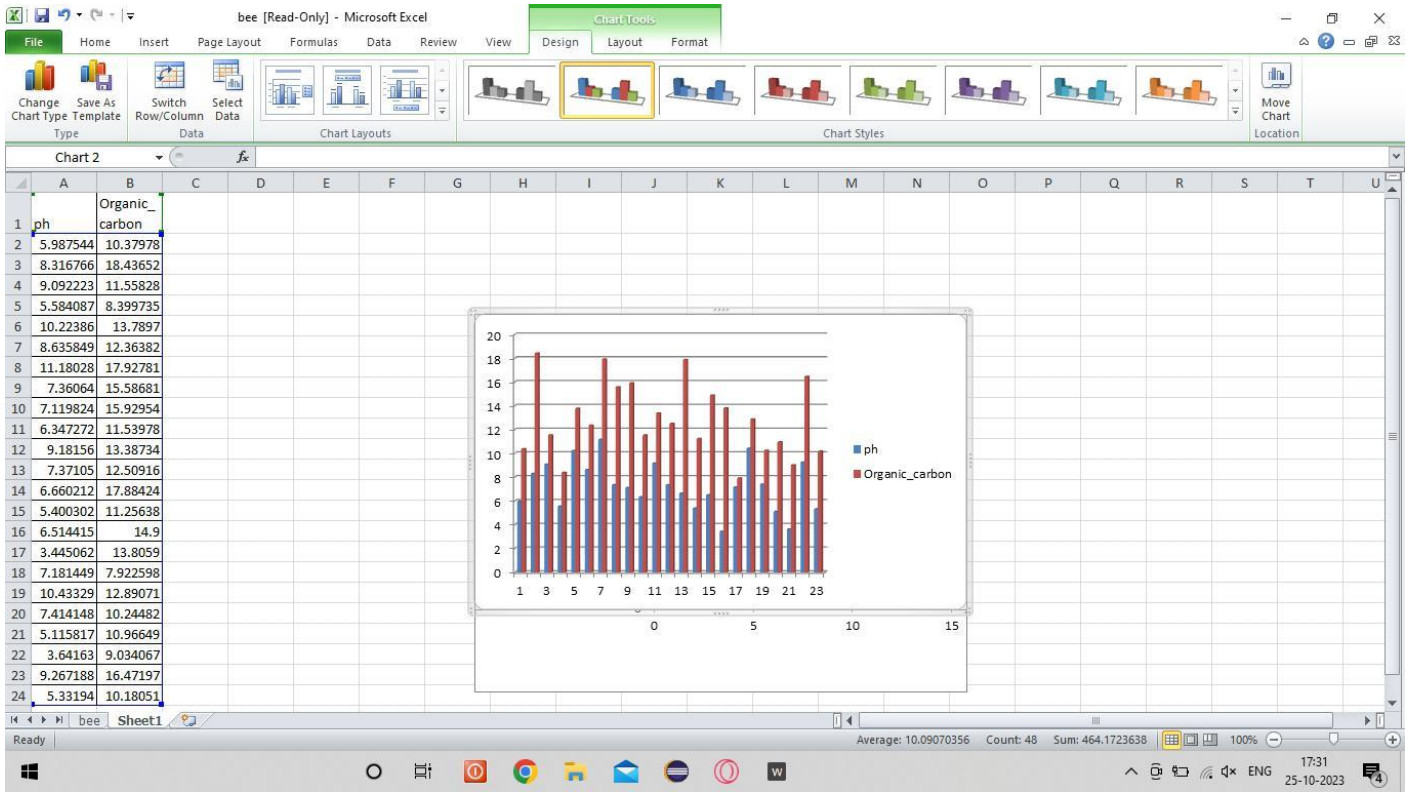
	A	B	C
1	ph	Organic_carbon	CORR
2	5.987544	10.37978	0.4394
3	8.316766	18.43652	0.434199
4	9.092223	11.55828	0.462734
5	5.584087	8.399735	0.435915
6	10.22386	13.7897	0.439364
7	8.635849	12.36382	0.456353
8	11.18028	17.92781	0.314748
9	7.36064	15.58681	0.304527
10	7.119824	15.92954	0.303653
11	6.347272	11.53978	0.301184
12	9.18156	13.38734	0.282441
13	7.37105	12.50916	0.281318
14	6.660212	17.88424	0.326352
15	5.400302	11.25638	0.321729
16	6.514415	14.9	0.355175
17	3.445062	13.8059	0.639498
18	7.181449	7.922598	0.758931
19	10.43329	12.89071	0.816123
20	7.414148	10.24482	0.922355
21	5.115817	10.96649	0.929892
22	3.64163	9.034067	0.941457
23	9.267188	16.47197	1
24	5.33194	10.18051	#DIV/0!

Histogram:



	A	B	C	D
1	04.890455	Frequency	04.890455	Frequency
2	117.7912	19	117.7912	19
3	140.7391	0	140.7391	0
4	156.705	0	156.705	0
5	165.5208	0	165.5208	0
6	168.2837	0	168.2837	0
7	181.1015	0	181.1015	0
8	186.7329	0	186.7329	0
9	188.3133	0	188.3133	0
10	198.7674	0	198.7674	0
11	203.3615	0	203.3615	0
12	207.9263	0	207.9263	0
13	209.6256	0	209.6256	0
14	214.3734	0	214.3734	0
15	214.4966	0	214.4966	0
16	227.2315	0	227.2315	0
17	235.0445	0	235.0445	0
18	248.0717	0	248.0717	0
19	273.8138	0	273.8138	0
20	More	0	More	0

Bar chart:


bee [Read-Only] - Microsoft Excel
Chart Tools
File
Home
Insert
Page Layout
Formulas
Data
Review
View
Design
Layout
Format
Change Chart Type
Save As Template
Type
Switch Row/Column
Select Data
Data
Chart Layouts
Chart Styles
Move Chart
Location
Chart 2
ph
Organic_carbon
5.987544 10.37978
8.316766 18.43652
9.092223 11.55828
5.584087 8.399735
10.22386 13.7897
8.635849 12.36382
11.18028 17.92781
7.36064 15.58681
7.119824 15.92954
6.347272 11.53978
9.18156 13.38734
7.37105 12.50916
6.660212 17.88424
5.400302 11.25638
6.514415 14.9
3.445062 13.8059
7.181449 7.922598
10.43329 12.89071
7.414148 10.24482
5.115817 10.96649
3.64163 9.034067
9.267188 16.47197
5.33194 10.18051
bee
Sheet1
Ready
Average: 10.09070356
Count: 48
Sum: 464.1723638
100%
ENG
17:31
25-10-2023

Predictive model:

- Predictive modeling is a commonly used statistical technique to predict future behavior. Predictive modeling solutions are a form of data-mining technology that works by analyzing historical and current data and generating a model to help predict future outcomes.

Linear regression:

- Linear regression is a statistical method for modeling relationships between a dependent variable with a given set of independent variables. Linear regression is a statistical method for modeling relationships between a dependent variable with a given set of independent variables. In this slide,we refer to dependent variables as responses and independent variables as features for simplicity.

Our predictive model: